


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)
**Search:** ☒ The ACM Digital Library ☐ The Guide

Searching within **The ACM Digital Library** with **Advanced Search**: (graph and thread and rule and static and intraprocedural) ([start a new search](#))

Found **16** of **250,316**

## REFINE YOUR SEARCH

[Search Results](#)
[Related Journals](#)
[Related SIGs](#)
[Related Conferences](#)

### ▼ Refine by Keywords

  
  
 Discovered Terms

### ▼ Refine by People

[Names](#)  
[Institutions](#)  
[Authors](#)  
[Reviewers](#)

### ▼ Refine by Publications

[Publication Year](#)  
[Publication Names](#)  
[ACM Publications](#)  
[All Publications](#)

### ▼ Refine by Conferences

[Sponsors](#)  
[Events](#)  
[Proceeding Series](#)

Results 1 - 16 of 16

Sort by  in

[Save results to a Binder](#)

#### 1 [Parametric shape analysis via 3-valued logic](#)

Mooly Sagiv, Thomas Reps, Reinhard Wilhelm

May 2002 **Transactions on Programming Languages and Systems (TOPLAS)**, Volume 3 Issue 3

**Publisher:** ACM [Request Permissions](#)

Full text available: Pdf (1.10 MB) [Additional Information: full citation, abstract, references, cited by, in terms](#)

**Bibliometrics:** Downloads (6 Weeks): 15, Downloads (12 Months): 123, Citation Count: 48

Shape analysis concerns the problem of determining "shape invariants" for programs perform destructive updating on dynamically allocated storage. This article presents a parametric framework for shape analysis that can be instantiated in different ...

**Keywords:** 3-valued logic, Abstract interpretation, alias analysis, constraint solving, destructive updating, pointer analysis, shape analysis, static analysis

#### 2 [Formalizing the safety of Java, the Java virtual machine, and Java card](#)

Pieter H. Hartel, Luc Moreau

December 2001 **Computing Surveys (CSUR)**, Volume 33 Issue 4

**Publisher:** ACM [Request Permissions](#)

Full text available: Pdf (442.86 KB) [Additional Information: full citation, abstract, references, cited by, in terms](#)

**Bibliometrics:** Downloads (6 Weeks): 65, Downloads (12 Months): 486, Citation Count: 13

We review the existing literature on Java safety, emphasizing formal approaches, and impact of Java safety on small footprint devices such as smartcards. The conclusion is that although a lot of good work has been done, a more concerted effort is ...

**Keywords:** Common criteria, programming

#### 3 [Korat: automated testing based on Java predicates](#)

Chandrasekhar Boyapati, Sariraz Khurshid, Darko Marinov

July 2002 **ISSTA '02: Proceedings of the 2002 ACM SIGSOFT international symposium on Software testing and analysis**

**Publisher:** ACM [Request Permissions](#)

Full text available: Pdf (171.43 KB) [Additional Information: full citation, abstract, references, cited by](#)

**Bibliometrics:** Downloads (6 Weeks): 16, Downloads (12 Months): 128, Citation Count: 70

This paper presents Korat, a novel framework for automated testing of Java program Given a formal specification for a method, Korat uses the method precondition to automatically generate all (nonisomorphic) test cases up to a given small size. Korat

## ADVANCED SEARCH

[Advanced Search](#)

## FEEDBACK


[Please provide us with feedback](#)

Found **16** of **250,316**

Also published in:


July 2002 **SIGSOFT Software Engineering Notes** Volume 27 Issue 4

#### 4 [Interprocedural compatibility analysis for static object preallocation](#)

 [Ovidiu Gheorghioiu, Alexandru Saiciu, Martin Rinard](#)

January 2003 **POPL '03**: Proceedings of the 30th ACM SIGPLAN-SIGACT symposium on Principles of programming languages

**Publisher:** ACM  [Request Permissions](#)

Full text available:  [Pdf](#) (277.65 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [in terms](#)

**Bibliometrics:** Downloads (6 Weeks): 2, Downloads (12 Months): 16, Citation Count: 6

We present an interprocedural and compositional algorithm for finding pairs of *compatible* allocation sites, which have the property that no object allocated at one site is live at same time as any object allocated at the other site. If an ...

**Keywords:** interprocedural analysis, memory preallocation, static analysis

Also published in:

January 2003 **SIGPLAN Notices** Volume 38 Issue 1

#### 5 [How to write system-specific, static checkers in metal](#)

 [Benjamin Chelf, Dawson Engler, Seth Haller](#)

January 2003 **PASTE '02**: Proceedings of the 2002 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering

**Publisher:** ACM  [Request Permissions](#)

Full text available:  [Pdf](#) (190.85 KB) Additional Information: [full citation](#), [references](#), [cited by](#), [index terms](#)

**Bibliometrics:** Downloads (6 Weeks): 2, Downloads (12 Months): 23, Citation Count: 4

Also published in:

January 2003 **SIGSOFT Software Engineering Notes** Volume 28 Issue 1

#### 6 [Pointer analysis for structured parallel programs](#)

 [Radu Rugina, Martin C. Rinard](#)

January 2003 **Transactions on Programming Languages and Systems (TOPLAS)**, Volume 25 Issue 1

**Publisher:** ACM  [Request Permissions](#)


Full text available:  [Pdf](#) (383.29 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

**Bibliometrics:** Downloads (6 Weeks): 12, Downloads (12 Months): 96, Citation Count: 3

This paper presents a novel interprocedural, flow-sensitive, and context-sensitive pointer analysis algorithm for multithreaded programs that may concurrently update shared pointers. The algorithm is designed to handle programs with structured parallel ...


**Keywords:** Pointer analysis

#### 7 [A fast approximate interprocedural analysis for speculative multithreading compilation](#)

 [Anasua Bhowmik, Manoj Franklin](#)

June 2003 **ICS '03**: Proceedings of the 17th annual international conference on Supercomputing

**Publisher:** ACM  [Request Permissions](#)

Full text available:  [Pdf](#) (192.97 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [in terms](#)

**Bibliometrics:** Downloads (6 Weeks): 2, Downloads (12 Months): 32, Citation Count: 3

Speculative multithreading (SpMT) promises to be very effective for parallelizing non numeric programs. Data dependences are perhaps the most important factor in the crucial step of deciding the thread boundaries, and SpMT compilers need to estimate

**Keywords:** interprocedural analysis, pointer analysis, speculative multithreading (SpMT), thread-level parallelism (TLP)

## 8 [Ensuring code safety without runtime checks for real-time control systems](#)



Suman Kowshik, Dinakar Dhurjati, Vikram Adve

October 2002 **CASES '02**: Proceedings of the 2002 international conference on Compilers, architecture, and synthesis for embedded systems

**Publisher:** ACM

Full text available: [Pdf](#) (127.10 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [in terms](#)

**Bibliometrics:** Downloads (6 Weeks): 9, Downloads (12 Months): 58, Citation Count: 8

This paper considers the problem of providing safe programming support and enabling secure online software upgrades for control software in real-time control systems. In systems, offline techniques for ensuring code safety are greatly preferable ...

**Keywords:** compiler, control, programming language, real-time, security, static analysis

## 9 [Stack allocation and synchronization optimizations for Java using escape analysis](#)



Jong-Deok Choi, Manish Gupta, Mauricio J. Serrano, Vugranam C. Sreedhar, Samuel P. Midkiff

November 2003 **Transactions on Programming Languages and Systems (TOPLAS)**

Volume 25 Issue 6

**Publisher:** ACM [Request Permissions](#)

Full text available: [Pdf](#) (632.85 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [in terms](#), [review](#)

**Bibliometrics:** Downloads (6 Weeks): 13, Downloads (12 Months): 111, Citation Count: 14

This article presents an *escape analysis* framework for Java to determine (1) if an object is not reachable after its method of creation returns, allowing the object to be allocated on the stack, and (2) if an object is reachable only from a single ...

**Keywords:** Connection graphs, escape analysis, points-to graph

## 10 [Alias annotations for program understanding](#)



Jonathan Aldrich, Valentin Kostadinov, Craig Chambers

November 2002 **OOPSLA '02**: Proceedings of the 17th ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications

**Publisher:** ACM [Request Permissions](#)

Full text available: [Pdf](#) (336.14 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [in terms](#)

**Bibliometrics:** Downloads (6 Weeks): 15, Downloads (12 Months): 89, Citation Count: 48

One of the primary challenges in building and evolving large object-oriented systems is understanding aliasing between objects. Unexpected aliasing can lead to broken invariants, mistaken assumptions, security holes, and surprising side effects, all ...

**Keywords:** aliasing, aliasjava, encapsulation, java, ownership types, type inference,

uniqueness

Also published in:


November 2002 **SIGPLAN Notices** Volume 37 Issue 11

## 11 [Data size optimizations for java programs](#)

 [C. Scott Ananian, Martin Rinard](#)

July 2003 **LCTES '03**: Proceedings of the 2003 ACM SIGPLAN conference on Language, compiler, and tool for embedded systems

**Publisher:** ACM  [Request Permissions](#)

Full text available:  [Pdf](#) (349.36 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [in terms](#)

**Bibliometrics:** Downloads (6 Weeks): 2, Downloads (12 Months): 39, Citation Count: 16


We present a set of techniques for reducing the memory consumption of object-oriented programs. These techniques include analysis algorithms and optimizations that use the results of these analyses to eliminate fields with constant values, reduce the ...

**Keywords:** bitwidth analysis, embedded systems, field externalization, field packing optimizations, static specialization

Also published in:


July 2003 **SIGPLAN Notices** Volume 38 Issue 7

## 12 [Flow-sensitive type qualifiers](#)

 [Jeffrey S. Foster, Tachio Terauchi, Alex Aiken](#)

June 2002 **PLDI '02**: Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation

**Publisher:** ACM  [Request Permissions](#)

Full text available:  [Pdf](#) (298.28 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [in terms](#)

**Bibliometrics:** Downloads (6 Weeks): 8, Downloads (12 Months): 56, Citation Count: 80


We present a system for extending standard type systems with flow-sensitive type qualifiers. Users annotate their programs with type qualifiers, and inference checks that the annotations are correct. In our system only the type qualifiers are modeled ...

**Keywords:** alias analysis, constraints, effect inference, flow-sensitivity, linux kernel, locking, restrict, type qualifiers, types

Also published in:


May 2002 **SIGPLAN Notices** Volume 37 Issue 5

## 13 [Deriving specialized program analyses for certifying component-client conformance](#)

 [G. Ramalingam, Alex Warshavsky, John Field, Deepak Goyal, Mooly Sagiv](#)

June 2002 **PLDI '02**: Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation

**Publisher:** ACM  [Request Permissions](#)

Full text available:  [Pdf](#) (158.44 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [in terms](#)

**Bibliometrics:** Downloads (6 Weeks): 4, Downloads (12 Months): 38, Citation Count: 13


We are concerned with the problem of statically *certifying* (verifying) whether the client of a software component conforms to the component's constraints for correct usage. We show how conformance certification can be efficiently carried out in ...

**Keywords:** abstract interpretation, model checking, predicate abstraction, software components, static analysis

Also published in:


May 2002 **SIGPLAN Notices** Volume 37 Issue 5

**14** [A system and language for building system-specific, static analyses](#)

 Seth Haillem, Benjamin Chell, Yichen Xie, Dawson Engler

June 2002 **PLDI '02**: Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation

**Publisher:** ACM  [Request Permissions](#)

Full text available:  Pdf (276.96 KB) [Additional Information: full citation, abstract, references, cited by, in terms](#)

**Bibliometrics:** Downloads (6 Weeks): 12, Downloads (12 Months): 79, Citation Count: 86

This paper presents a novel approach to bug-finding analysis and an implementation that approach. Our goal is to find as many serious bugs as possible. To do so, we designed a flexible, easy-to-use extension language for specifying analyses and an ..

**Keywords:** error detection, extensible compilation

Also published in:


May 2002 **SIGPLAN Notices** Volume 37 Issue 5

**15** [Using types to analyze and optimize object-oriented programs](#)

 Amer Diwan, Kathryn S. McKinley, J. Eliot B. Moss

January 2001 **Transactions on Programming Languages and Systems (TOPLAS)**, Volume 23 Issue 1

**Publisher:** ACM  [Request Permissions](#)


Full text available:  Pdf (414.51 KB) [Additional Information: full citation, abstract, references, cited by, in terms](#)

**Bibliometrics:** Downloads (6 Weeks): 11, Downloads (12 Months): 107, Citation Count: 10

Object-oriented programming languages provide many software engineering benefits these often come at a performance cost. Object-oriented programs make extensive use of method invocations and pointer dereferences, both of which are potentially costly ...


**Keywords:** alias analysis, classes and objects, method invocation, object orientation polymorphism, redundancy elimination

**16** [Using indexed data structures for program specialization](#)

 Jung Gyu Park, Myong-Soon Park

September 2002 **ASI A-PEPM '02**: Proceedings of the ASIAN symposium on Partial evaluation and semantics-based program manipulation

**Publisher:** ACM  [Request Permissions](#)

Full text available:  Pdf (201.23 KB) [Additional Information: full citation, abstract, references, cited by, in terms](#)





**Bibliometrics:** Downloads (6 Weeks): 1, Downloads (12 Months): 11, Citation Count: 1

Given a program and values of static (fixed) inputs, program specialization generates an optimized version of the program that only requires dynamic (run-time) inputs. It has been a useful tool for such areas as operating systems, multimedia applications, ...

**Keywords:** automated software engineering, partial evaluation, program specialization, staged computation optimization

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2009 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)